Bagging & Boosting

Session Outline

- Ensemble Techniques
- Concept of Bias-Variance Trade Off
- How boosting reduces over all error
- When to use boosting- Examples of algorithms that use boosting
- Bagging-Use of Bagging to reduce over all error
- When to use Bagging-Examples of algorithms that uses Bagging

Concept of Bias-Variance Trade Off

- Overfitting
- Underfitting
- Bias
- Variance

Bias & Variance

Bias

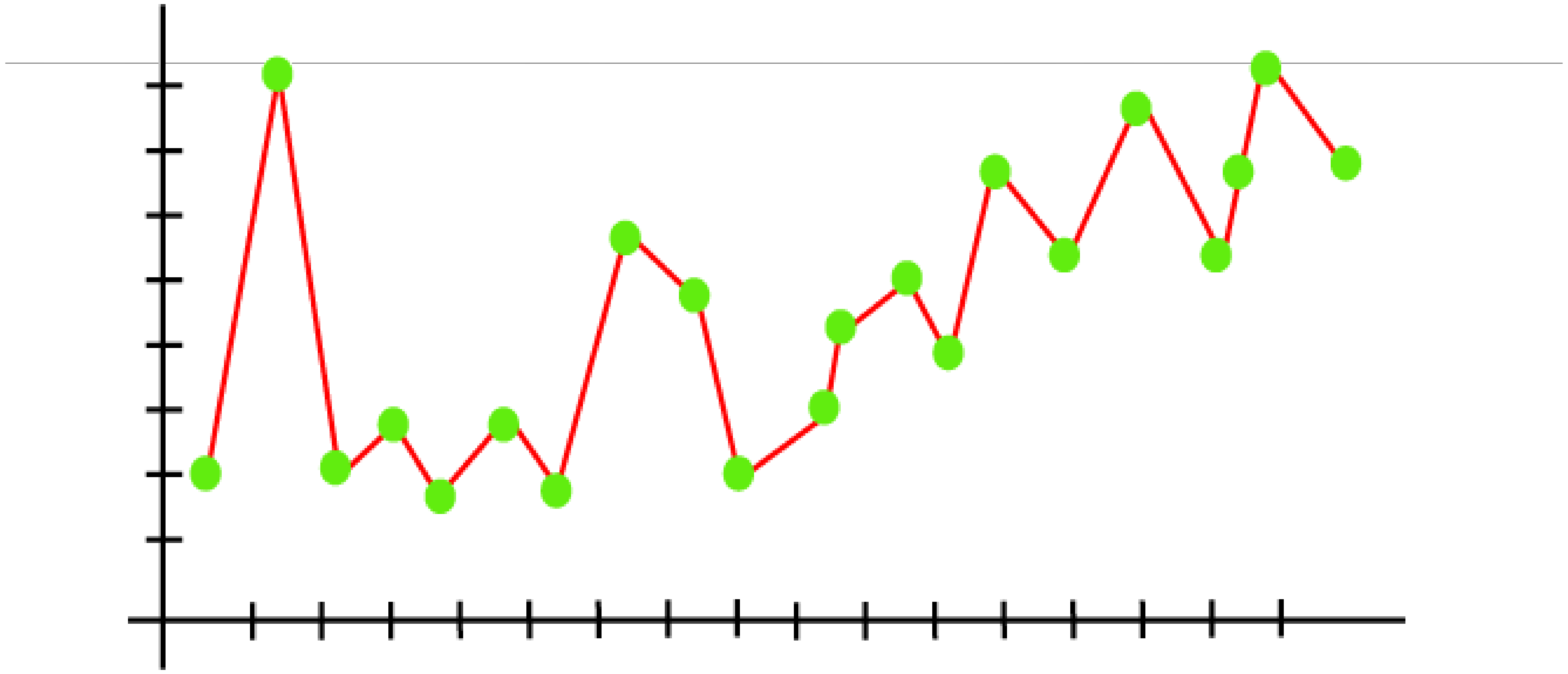
- It is simply how far our predicted value is with respect to the actual value.
- We have two different types in bias, they are:
- Low Bias: Suggests less far from the actual target value
- High-Bias: Suggests more far from the actual target value.

Variance

- Variance means when a model performs well on train data during training and does not generalize on the new data.
- It is simply the error rate of the test data.
- How much it is varying the performance/accuracy on training and testing.
- We have two different types of invariance, they are:
- Low variance: shows less difference in test accuracy with respect to train accuracy.
- High-variance: shows a high difference in test accuracy with respect to train accuracy.

Overfitting

- Overfitting occurs when our machine learning model tries to cover all the data points or more than the required data points present in the given dataset.
- Because of this, the model starts caching noise and inaccurate values present in the dataset, and all these factors reduce the efficiency and accuracy of the model.
- Overfitting is the main problem that occurs in supervised learning.
- The overfitted model has low bias and high variance.

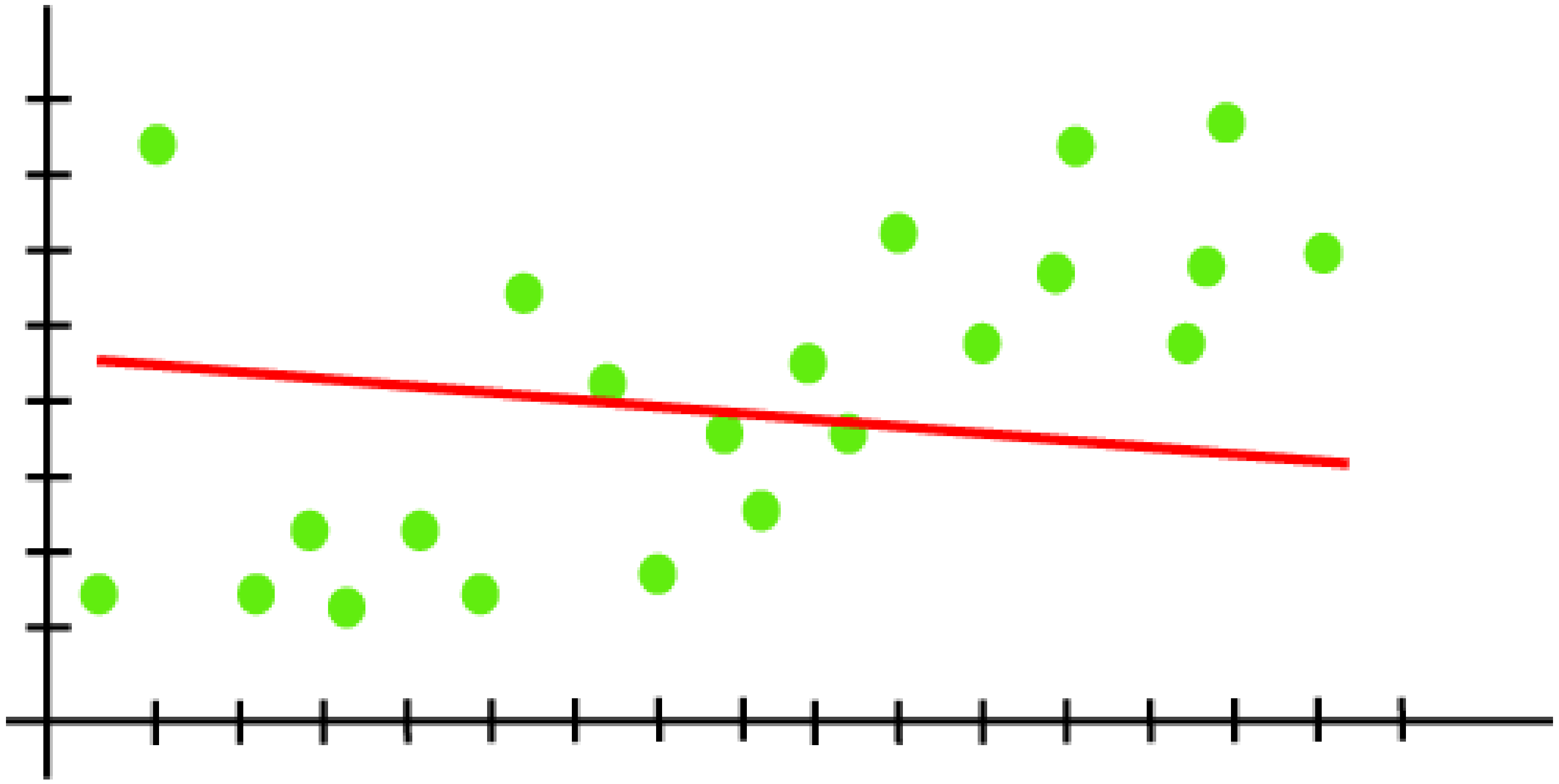


How to avoid the Overfitting in Model

- Cross-Validation
- Training with more data
- Removing features
- Early stopping the training
- Regularization
- **Ensembling**

Underfitting

- Underfitting occurs when our machine learning model is not able to capture the underlying trend of the data.
- the model is not able to learn enough from the training data, and hence it reduces the accuracy and produces unreliable predictions.
- An underfitted model has high bias and low variance

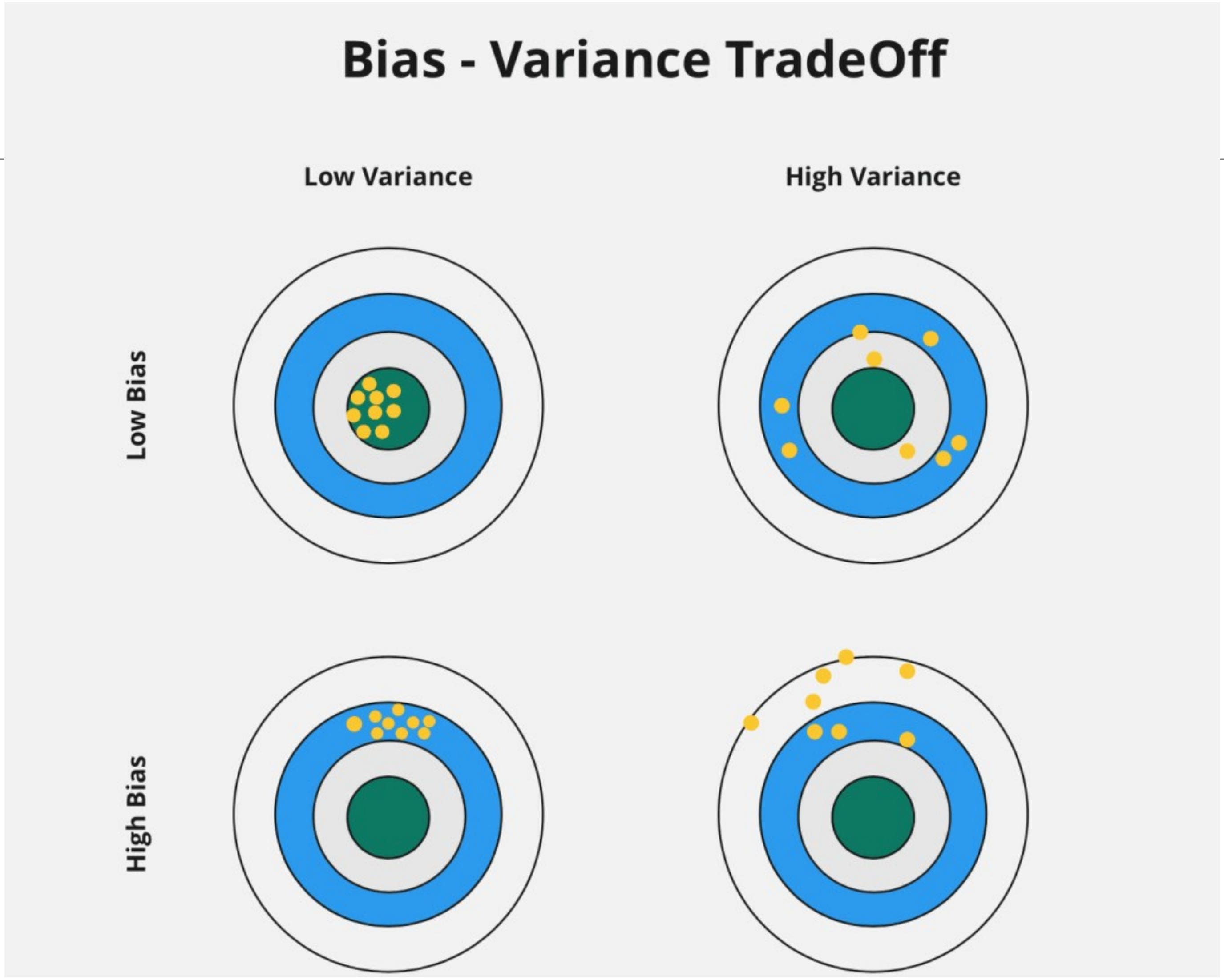


How to avoid underfitting:

- By increasing the training time of the model.
- By increasing the number of features.

Bias variance tradeoff

- Finding the right balance between bias and variance of the model is called the Bias-variance tradeoff.
- If our model is too simple and has very few parameters then it may have high bias and low variance.
- On the other hand, if our model has a large number of parameters then it’s going to have high variance and low bias.
- So we need to find a good balance without overfitting and underfitting the data.

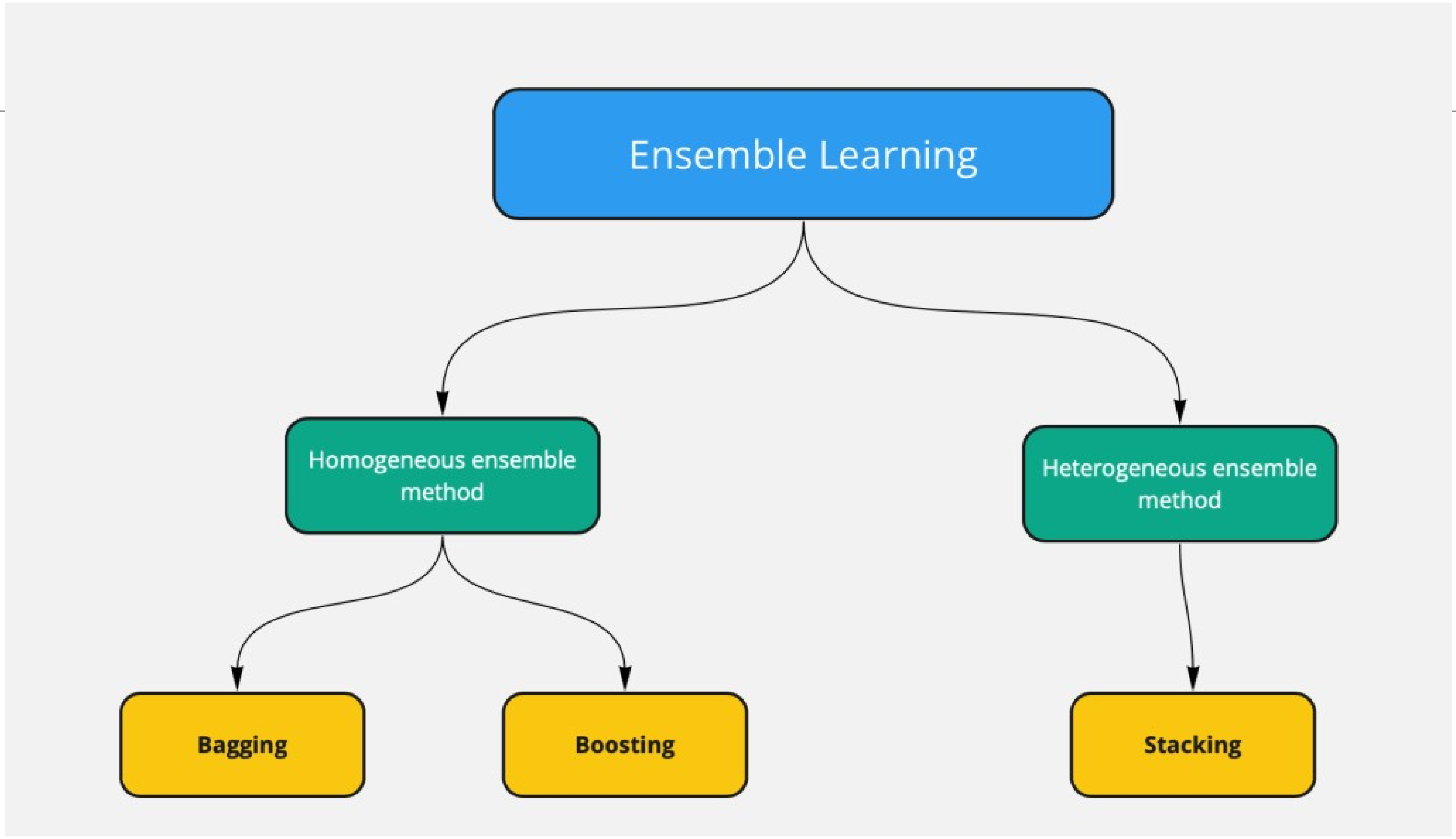

Bias - Variance TradeOff
Low Variance
High Variance
Low Bias
High Bias

From the diagram we have to know a few things;

Low bias & Low variance -------> Good model
Low bias & High Variance -------> Overfitted model
High bias & Low variance ------> Under fitted model

Ensembling

- Ensembling refers to the process of combining the predictions of multiple models to create a more accurate and robust prediction.
- Ensembling can be a very effective way to improve the accuracy of machine learning models.
- The goal of ensembling is to reduce the variance and bias of individual models, and to improve the overall prediction performance.
- Ensemble techniques can be applied to a wide range of machine learning problems, including classification, regression, and clustering.
-


Ensemble Learning
Homogeneous ensemble method
Heterogeneous ensemble method
Bagging
Boosting
Stacking

Inclusys Org Foundation
Empowering the Neurodivergent

Ensemble methods are of two types:

- Homogeneous ensemble methods
- Heterogeneous ensemble methods

- **<u>Homogeneous ensemble methods</u>**
- In homogeneous ensemble methods all the individual models are built using the same machine learning algorithm.
- Even if we are using the same training data to build the same machine learning algorithm, still all the models will be different.
- The individual models are called weak learners.
- For example, if the individual model is a decision tree then one good example for the ensemble method is random forest.

Heterogeneous Ensemble Method

- Each model will be different but uses the same training data.
- Here also the individual models are called weak learners.
- The stacking method will fall under the heterogeneous ensemble method.

Popular ensemble techniques

- Bagging
- Boostng
- Random Forest
- Stacking

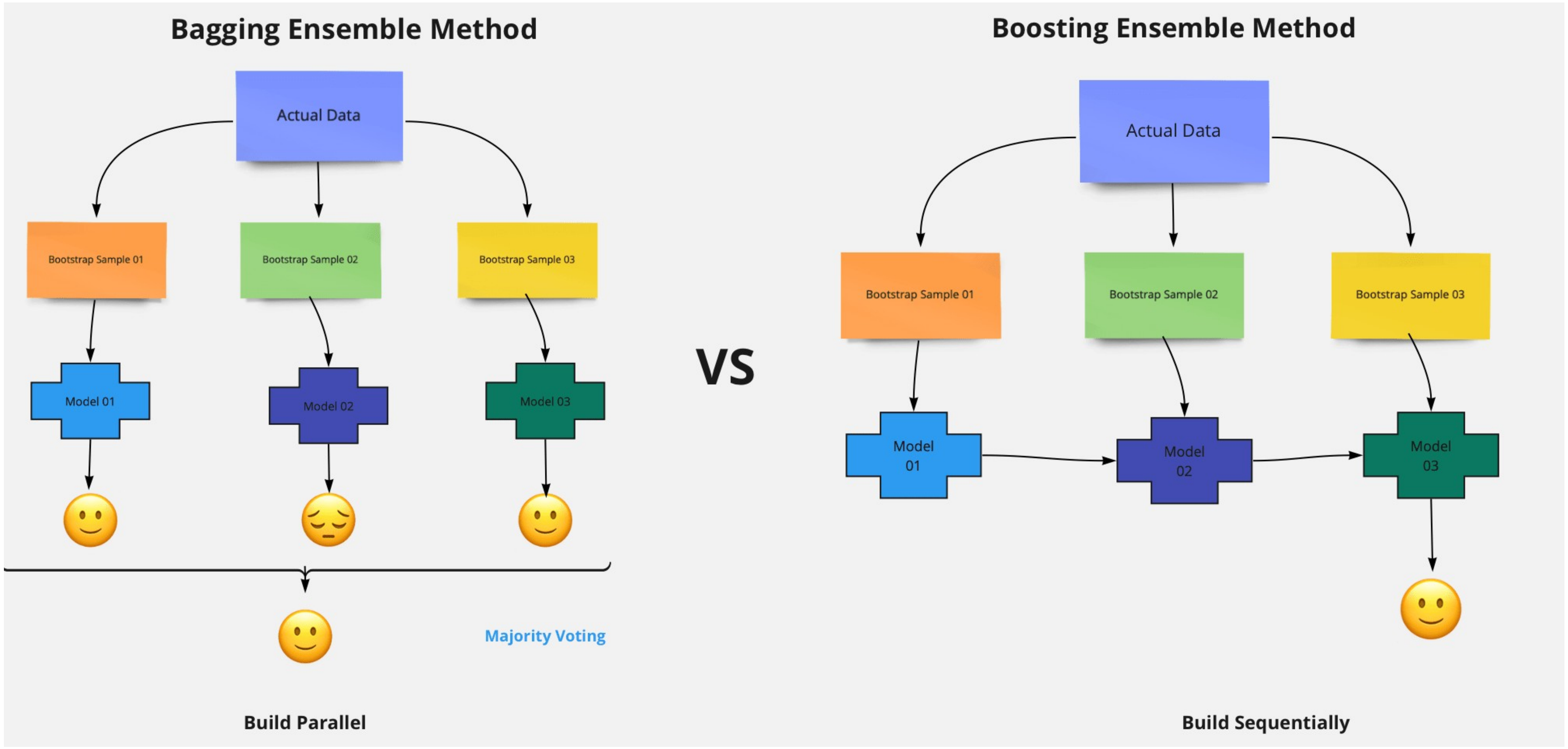
Bagging

- Bagging stands for Bootstrap Aggregating.
- This technique involves training multiple models on different subsets of the training data, and then aggregating their predictions to make the final prediction.
- Bagging can help to reduce overfitting and improve the stability of the model.

- This technique involves training multiple models on different subsets of the training data, and then aggregating their predictions to make the final prediction.
- Bagging can help to reduce overfitting and improve the stability of the model.

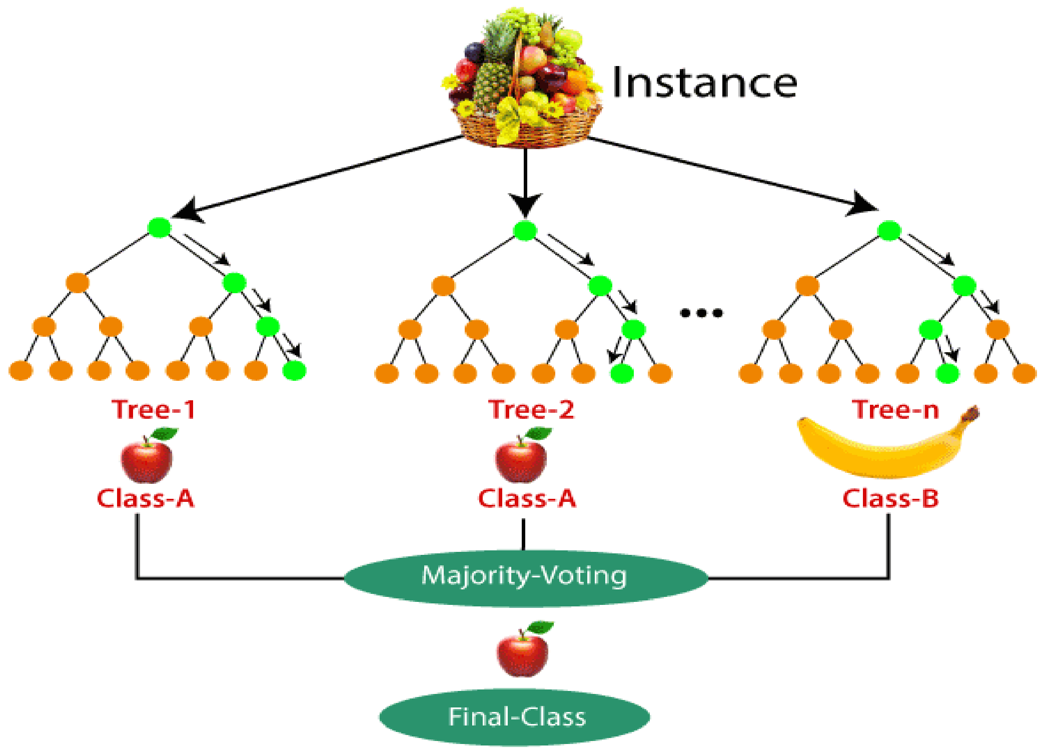
Boostning

- Boosting involves training multiple models in sequence, with each subsequent model trying to correct the errors of the previous model.
- Boosting can be particularly effective for improving the performance of weak learners.


Bagging Ensemble Method
Actual Data
Bootstrap Sample 01
Bootstrap Sample 02
Bootstrap Sample 03
Model 01
Model 02
Model 03
Majority Voting
Build Parallel
VS
Boosting Ensemble Method
Actual Data
Bootstrap Sample 01
Bootstrap Sample 02
Bootstrap Sample 03
Model 01
Model 02
Model 03
Build Sequentially

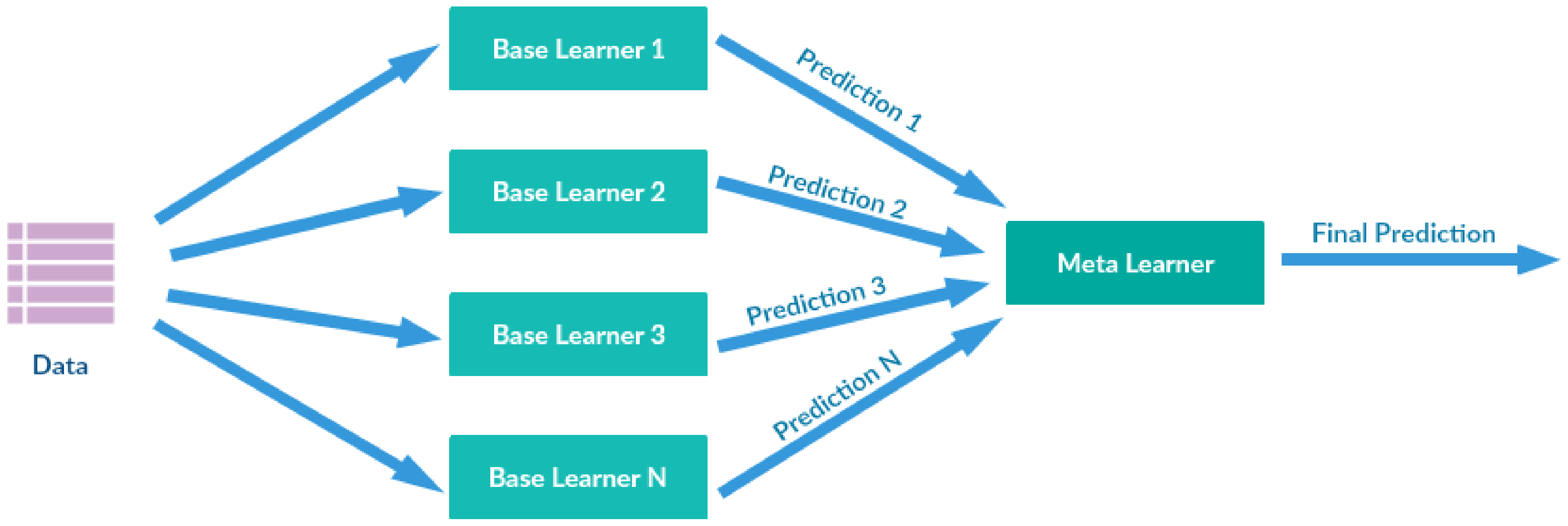
Random Forest

- Random Forest is an extension of bagging that uses a set of decision trees to make predictions.
- Random Forest can handle both categorical and continuous variables, and can help to reduce the risk of overfitting.
-

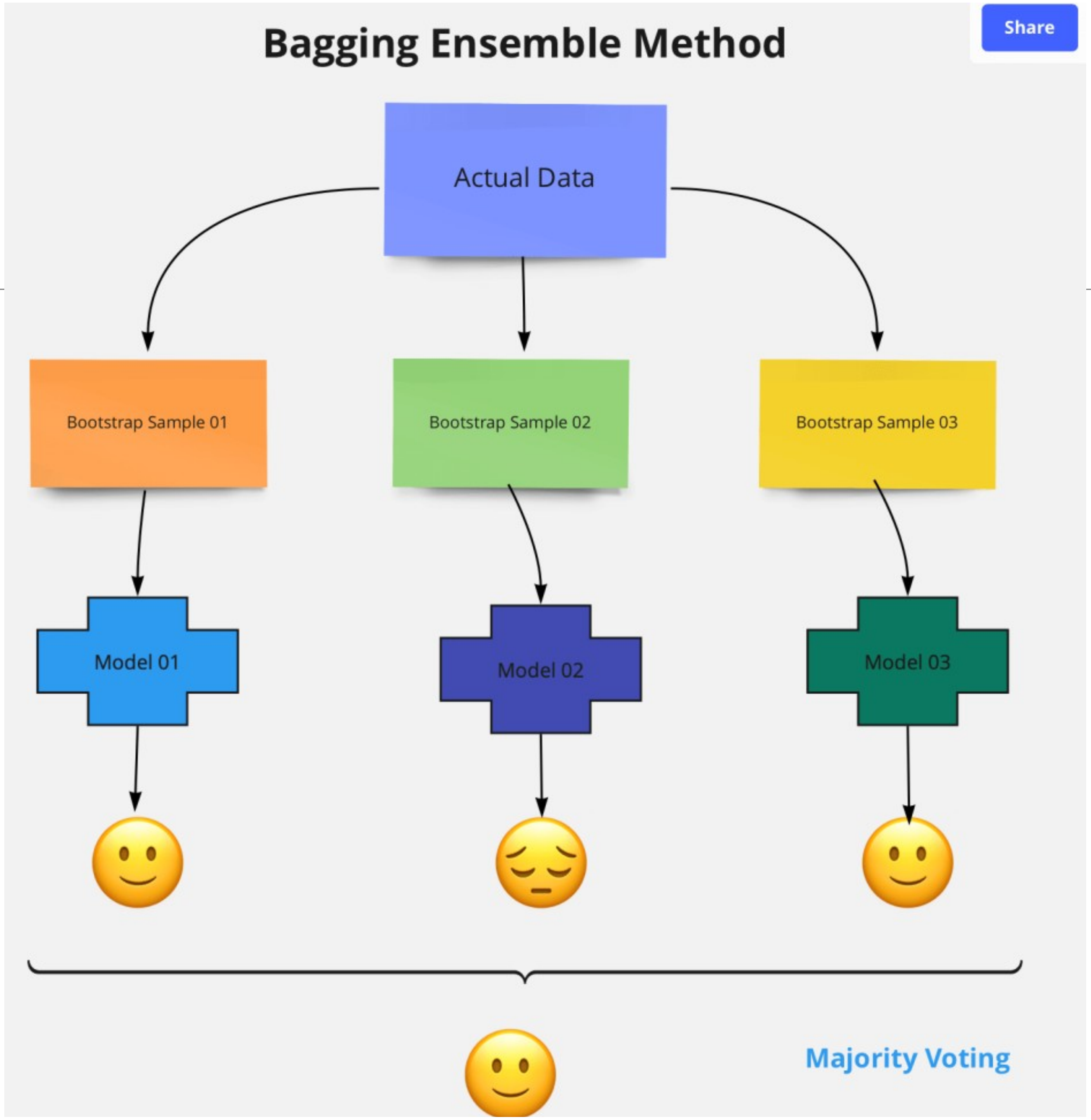


Stacking

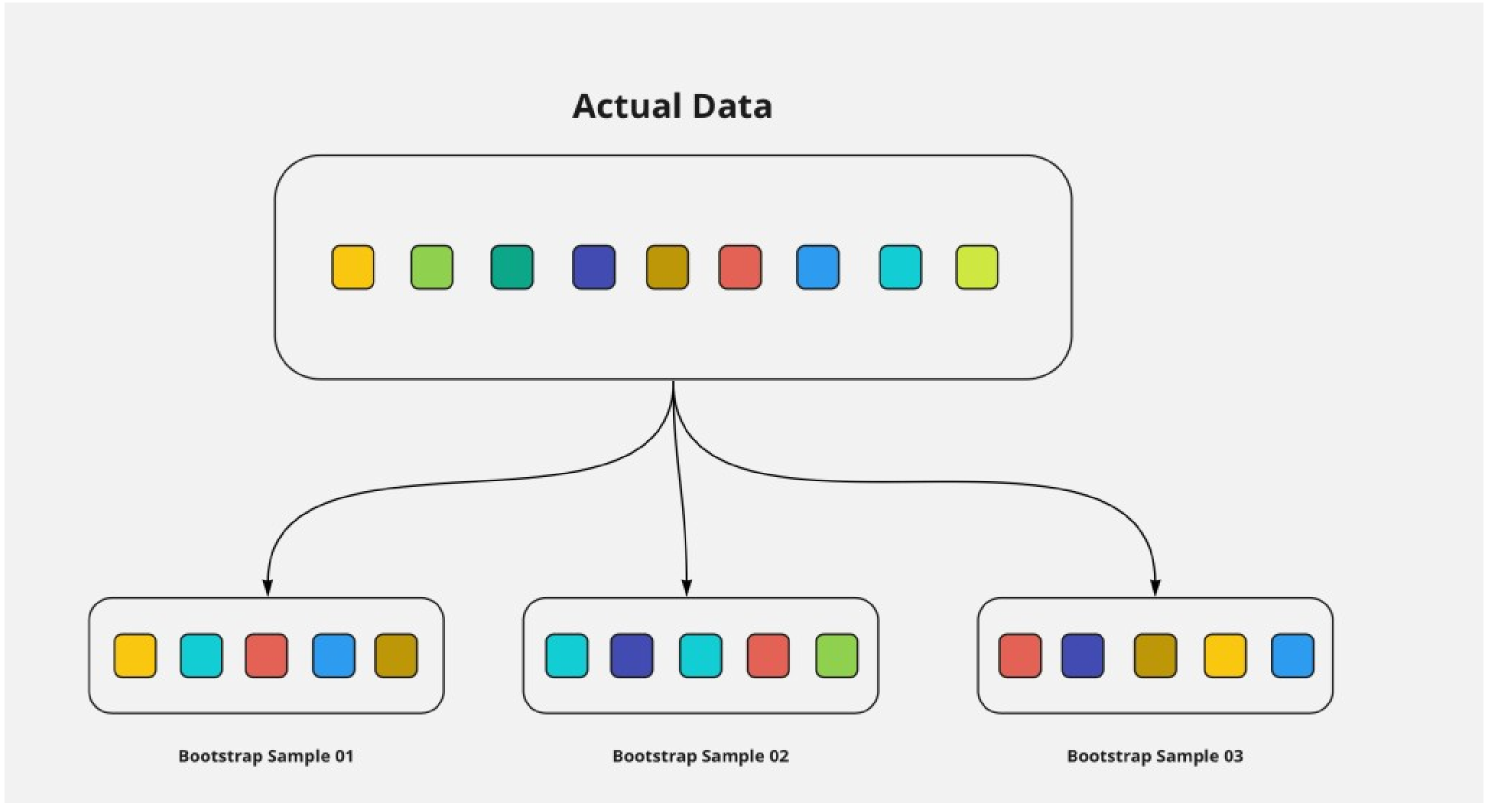
- Stacking involves training multiple models on the same training data, and then using a meta-model to combine their predictions.
- Stacking can help to combine the strengths of different models and produce a more accurate prediction.


Data
Base Learner 1
Base Learner 2
Base Learner 3
Base Learner N
Prediction 1
Prediction 2
Prediction 3
Prediction N
Meta Learner
Final Prediction

Bagging


Bagging Ensemble Method
Share
Actual Data
Bootstrap Sample 01
Bootstrap Sample 02
Bootstrap Sample 03
Model 01
Model 02
Model 03
Majority Voting

Bootstrapping



- Bootstrapping is a statistical method to create sample data without leaving the properties of the actual dataset.
- The individual samples of data called bootstrap samples.
- Each sample is an approximation for the actual data.
- These individual sample has to capture the underlying complexity of the actual data.
- All data points in the samples are randomly taken with replacement.

- In the above image from the actual dataset, we created 3 bootstrap samples. In this case, we are creating an equal size sample. We don’t have any hard rule saying all the bootstrap sample sizes should be the same.
- In the bootstrapping properties, we said the data points will take randomly and with replacement. From the above image, the second bootstrapping sample is having a repeated data point (which is light green.)

When to use Bagging

Here are some scenarios where bagging can be useful:

- **<u>High variance models</u>**
- Bagging can be useful for models that have high variance, such as decision trees or random forests.
- By creating multiple models on different subsamples of the training data, bagging can reduce the variance and improve the accuracy of the model.
- **<u>Large datasets</u>**
- Bagging can also be useful for large datasets where training a single model can be time-consuming.
- By creating multiple models in parallel on different subsamples of the data, bagging can speed up the training process.

- **Noisy datasets**
- Bagging can also be useful for datasets that are noisy or contain outliers.
- By training multiple models on different subsamples of the data, bagging can reduce the impact of the noise and improve the robustness of the model.

- **Ensemble methods**
- Bagging can be used as a component of ensemble methods, such as random forests or stacked ensembles.
- In these cases, bagging is used to reduce the variance of the individual models, and the ensemble method is used to combine the predictions of the individual models into a stronger model.

Examples of algorithms that uses Bagging

- **Random Forest**
- Random forest is an ensemble learning algorithm that uses bagging to reduce the variance of decision trees.
- In each iteration, a random subset of the features is selected for splitting, and a decision tree is trained on a subsample of the training data.
- The final prediction is then made by combining the predictions of all the decision trees.

- **Bagged Decision Trees**
- Bagging can also be used with individual decision trees to reduce the variance of the model.
- In each iteration, a random subsample of the training data is selected, and a decision tree is trained on the subsample.
- The final prediction is then made by combining the predictions of all the decision trees.

Bagging with K-Nearest Neighbors

- Bagging can also be used with K-Nearest Neighbors (KNN) to improve the robustness of the model.
- In each iteration, a random subsample of the training data is selected, and a KNN model is trained on the subsample.
- The final prediction is then made by combining the predictions of all the KNN models.

Bagging with Support Vector Machines

- Bagging can also be used with Support Vector Machines (SVMs) to improve the robustness of the model.

- In each iteration, a random subsample of the training data is selected, and an SVM model is trained on the subsample. The final prediction is then made by combining the predictions of all the SVM models.

<u>Use of Bagging to reduce over all error</u>

Reducing variance

- Bagging can reduce the variance of a model by training multiple models on different subsamples of the training data.
- Each model will have a different set of training samples, which will lead to differences in the predictions.
- By combining the predictions of the individual models, bagging can reduce the variance and improve the accuracy of the model.

Reducing overfitting:

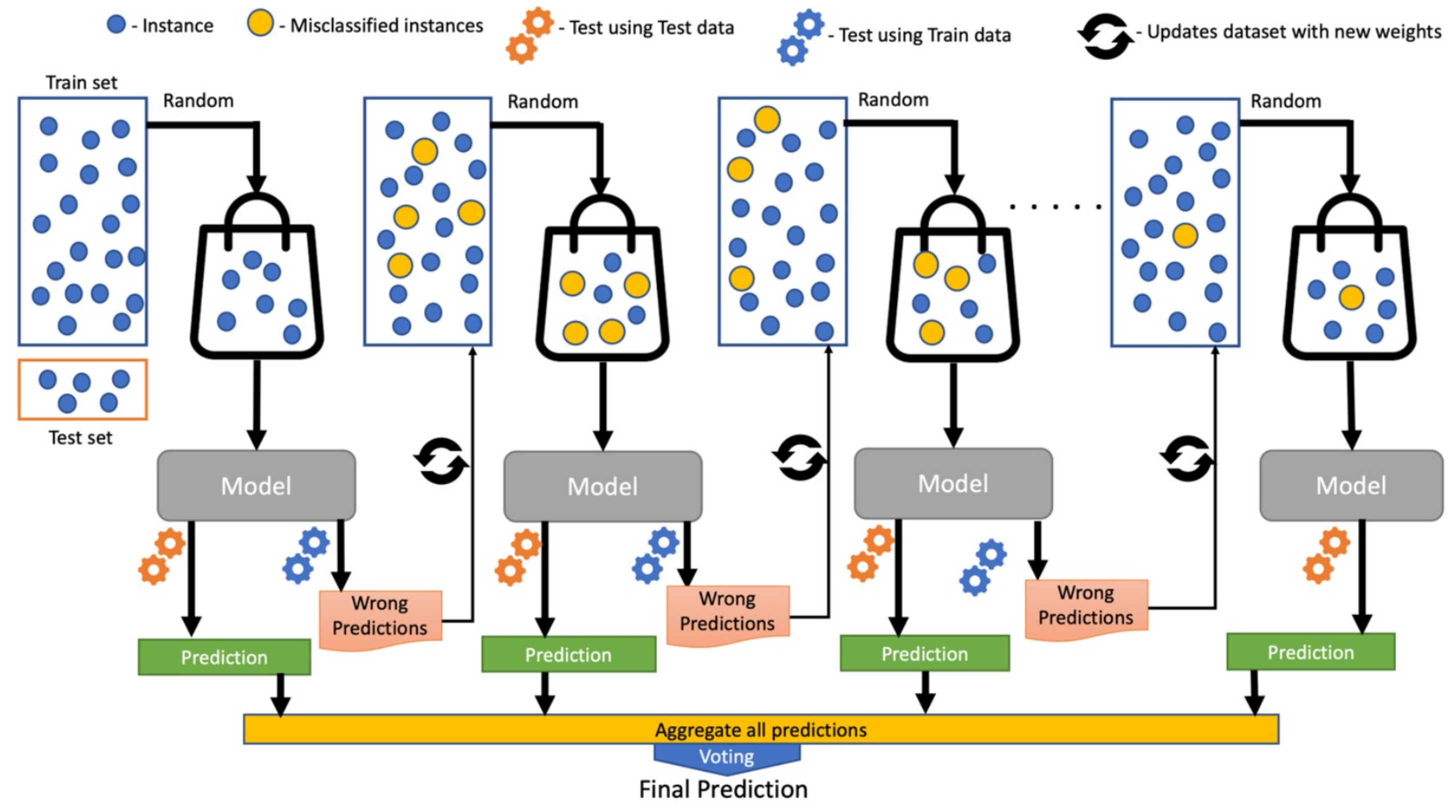
- Bagging can also reduce the risk of overfitting by creating multiple models with different subsets of the training data.
- Overfitting occurs when a model is too complex and fits the noise in the data, rather than the underlying patterns.

- By training multiple models on different subsamples of the data, bagging can create a more generalized model that is less likely to overfit the data.

<u>Robustness to outliers and noise</u>

- Bagging can also improve the robustness of a model to outliers and noisy data.
- By training multiple models on different subsamples of the data, bagging can reduce the impact of outliers and noisy data on the final prediction.

Boosting


- Instance
- Misclassified instances
- Test using Test data
- Test using Train data
- Updates dataset with new weights
Train set
Test set
Random
Random
Random
Random
Model
Model
Model
Model
Wrong Predictions
Wrong Predictions
Wrong Predictions
Prediction
Prediction
Prediction
Prediction
Aggregate all predictions
Voting
Final Prediction

When to use boosting

- Boosting is a machine learning technique that can be used in a variety of situations.
- In general, boosting is useful when you have a large number of *weak learners* (models that are only slightly better than random guessing) and you want to combine them to create a more powerful model.
- Boosting is often used in the following situations:
- Classification problems:

➢ Boosting can be used to improve the accuracy of classification models.

➢ For example, boosting is often used in the context of binary classification problems, where the goal is to predict one of two possible outcomes (e.g., whether a customer will buy a product or not).

Regression problems

- Boosting can also be used to improve the accuracy of regression models.
- For example, boosting can be used to predict the value of a continuous variable, such as the price of a house or the age of a person.

Large datasets

- Boosting is particularly useful when you have a large dataset.
- This is because boosting can be used to train multiple models in parallel, which can significantly reduce the time required to train the model.

Imbalanced datasets:

- Boosting can be used to improve the accuracy of models on imbalanced datasets.
- For example, if you have a dataset where one class is much more common than the other, boosting can be used to improve the accuracy of the model on the minority class.

Overall, boosting is a versatile and powerful technique that can be used in a wide range of machine learning problems. However, it may not be suitable for every situation, and it is important to evaluate whether boosting is the right approach for your specific problem.

Examples of algorithms that use boosting

Boosting is a popular ensemble learning technique that can be applied to a variety of machine learning algorithms. Here are some examples of machine learning algorithms that use boosting:

AdaBoost

- Adaptive Boosting, or AdaBoost for short, is a boosting algorithm that uses a set of weak learners (typically decision trees with a single split) to iteratively train a stronger classifier.
- In each iteration, the algorithm assigns weights to the misclassified samples, and the subsequent weak learner focuses on correctly classifying the misclassified samples.

Gradient Boosting

- Gradient Boosting is a boosting algorithm that optimizes a differentiable loss function by iteratively adding weak learners.
- In each iteration, the algorithm adds a new weak learner that is trained to minimize the gradient of the loss function with respect to the predicted values. Gradient Boosting is commonly used for regression and classification problems.

XGBoost

- Extreme Gradient Boosting, or XGBoost, is a gradient boosting algorithm that uses a more advanced implementation of the gradient descent algorithm, as well as regularization techniques to prevent overfitting.
- XGBoost has become a popular choice for a wide range of machine learning tasks, including classification, regression, and ranking.

Inclusys Org Foundation
Empowering the Neurodivergent

LightGBM

- Light Gradient Boosting Machine, or LightGBM, is a gradient boosting algorithm that is designed to handle large-scale datasets.
- It uses a leaf-wise growth strategy to reduce the number of decision trees needed, and it can handle categorical features directly without requiring one-hot encoding.

`

- **<u>CatBoost:</u>**
- CatBoost is a gradient boosting algorithm that is designed to handle categorical features and high-cardinality data.
- It uses a combination of ordered boosting and random permutations to handle categorical features, and it also includes several regularization techniques to prevent overfitting.

How boosting reduces over all error

- **Boosting reduces overall error by addressing two types of errors: bias and variance**.
- Bias is the error that comes from a model being too simple to capture the underlying patterns in the data, while variance is the error that comes from a model being too complex and fitting the noise in the data.
- Boosting reduces bias by iteratively adding new weak learners that can capture the patterns missed by the previous learners.
- This increases the complexity of the model and reduces the bias.
- Boosting reduces variance by combining the predictions of multiple weak learners, which helps to reduce the impact of the noise in the data.

Another way that boosting reduces overall error is by addressing the problem of overfitting.

- Overfitting occurs when a model is too complex and fits the noise in the data, rather than the underlying patterns.
- Boosting helps to reduce overfitting by using weak learners that are less complex and more likely to generalize well to new data.
- Additionally, boosting includes techniques such as early stopping and regularization to prevent overfitting.

Inclusys Org Foundation
Empowering the Neurodivergent

Inclusys Org Foundation
Empowering the Neurodivergent

Thank You